Pergamon

Computers Math. Applic. Vol. 29, No. 5, pp. 51–63, 1995

0898-1221/95 $9.50 + 0.00

0898-1221(94)00248-7

Semantic Relatedness

J. A. JOHNSON
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2

(Received April 1994; accepted June 1994)

Abstract—The notions of n-ary set and min/max value (a generalization of total/partial and into/onto mapping) are defined for conceptual modeling for the design of a natural language interface. SET-Association diagrams (a generalization of Entity Relationship diagrams) are used to formalize the intuitive notion of semantic relatedness.

Keywords—SET model, Parallelism, Equivalent SET schemas, Intensional equality.

1. INTRODUCTION

Intuitively, *semantic relatedness* between concepts in the domain is the likelihood that they will be used together in a sentence or dialog. Previous semantic relatedness measures [1–5] equate the likelihood that concepts will be used together with the number of concepts connecting them. For example, in the music domain, the concepts of *tone* and *pitch* are strongly related because there is a direct link between them. The concept of *pitch* refers to the highness or lowness of tone. Concepts may be strongly related in one domain and only weakly related (or not related at all) in another. In a baseball domain, there is a circuitous and unusual path of concepts connecting those of *pitch* and *tone*. Perhaps good muscle tone improves a baseball player's pitching arm, but this is an unlikely way of relating the two concepts. Hence, in the baseball domain, the concepts of *pitch* and *tone* are weakly related. Synonyms are strongly semantically related. Since they denote the same concept, there are no intervening concepts required to relate them. In the music domain, the nouns "pitch," "quality," and "strength" are synonymous, but in the baseball domain we are hard pressed to find a connection between them.

In work by Wald and Sorenson [6], a heuristic is provided for solving the query inference problem. Johnson and Rosenberg [7] extend Wald and Sorenson's heuristic to provide a measure of semantic relatedness for resolving ambiguities in natural language database requests. Bear and Hobbs [8] show a method of representing prepositional phrase attachment ambiguities in one expression motivated by the desire for efficient resolution of the ambiguity by means of parallel processing. Separate expressions for each possible attachment alternative results in duplication of processing if each expression must be evaluated independently. The expressions differ little from each other except in the attachment of the prepositional phrase. Concurrent representation of the attachment alternatives and parallel processing to choose the best attachment site is a more efficient strategy.

Schmalz [9] shows that the processes involved in parsing, reference resolution, and inference can be mapped to a particular algebra [10] suitable for efficient implementation on parallel ma-

Special thanks to L. Saxton and N. Cercone for valuable comments and suggestions on an earlier version of this manuscript. I would also like to acknowledge the financial support of the National Sciences and Engineering Research Council of Canada.

chines. Although this work has practical significance in terms of processing efficiency, it also has a theoretical impact for the design of natural language interfaces. The mapping of processes involved in natural language understanding to a rigorous notation with algebraic properties provides a framework with predictive capability. For example, if two English inputs are equivalent in the sense that they are rephrasings of each other, then operations in the algebra should permit translations between their corresponding internal representations.

Semantic relatedness is useful for resolution of ambiguities in natural language. Possible interpretations for a natural language input are partially ordered by their semantic relatedness. Different interpretations may require different numbers of concepts to relate the concepts denoted by words in the natural language input. Semantic relatedness for parallel representation of different interpretations of the same natural language input has not been addressed in the research so far. Parallel internal representations of Bear and Hobbs need to be generalized to ambiguities additional to those resulting from prepositional phrase attachment alternatives such as, for example, those resulting from word sense ambiguities.

The semantic relatedness measure (SRM) of Johnson and Rosenberg uses a variant of Entity Relationship diagram called an SET-Association diagram. Informally, semantic relatedness for an interpretation (possible reading) of a natural language database request is the strength of the relationship or the cohesiveness that the interpretation gives to the entities referred to in the request. Semantic relatedness is computed for each interpretation of a request which results in an ordering on the interpretations from most likely to least likely. The interpretation that most strongly relates the entities referred to in the request is considered to be the one that is most likely intended by the person who formulated the request. Parallel SET-Association diagrams should provide more efficient processing (like Bear and Hobbs and Schmalz) for resolving ambiguities.

In this paper, we provide the conceptual foundations needed to express the informal notion of semantic relatedness. Parallelism is part of the foundations needed. With these conceptual foundations, we will be able to translate between alternate rephrasings of the same English input.

2. DEFINITIONS AND CONCEPTUAL FRAMEWORK

The SET Conceptual model [11,12] is a refinement of the Entity Relationship (ER) model based on the set theories of Gilmore [13]. Extending the work of Johnson and Rosenberg [7] in which the SET model is used in the design of a natural language interface, we introduce parallel SET schemas as a way of avoiding ambiguities in natural language and completely characterizing the notion of semantic relatedness. The *intension* of a set is a statement of a property (possibly complex) that determines membership in the set expressed in either a formal (e.g., the relational algebra) or an informal (e.g., English) language. The *extension* of a set is the membership of the set (the collection of entities that satisfy the property expressed by the intension of the set). For example, $\{x \in Z : 3 \leq x \leq 5\}$ is the intension of a set whose extension is $\{3, 4, 5\}$.

DEFINITION 1 [11]. *An n-ary association S is a subset of the Cartesian product $(S_1 \times \cdots \times S_n)$ of not necessarily distinct sets $S_1, \ldots, S_n$.*

DEFINITION 2 [11]. *The arity of an n-ary association is n.*

DEFINITION 3 [7]. *Each of the sets $S_1, \ldots, S_n$ is called a parent set of S and each may itself be an association.*

For example, the *Manager* association is a binary association which is a subset of the Cartesian product (*Employee* $\times$ *Employee*) where *Employee* is a set of employees. The *Manager* association has one distinct set *Employee* which plays a role as two different parent sets. An ordered pair with left element a and right element b will be denoted using angular brackets $\langle a, b \rangle$.

DEFINITION 4 [14]. *Given n-ary association $S \subseteq (S_1 \times \cdots \times S_n)$, an association entity is a tuple $\langle s_1, \ldots, s_n \rangle \in S$.*

DEFINITION 5 [14]. *An entity $e \in S_i$ participates in association entity $\langle s_1, \ldots, s_n \rangle \in S$ as the i^{th} component if and only if $e = s_i$.*

For example, if both a and b are members of *Employee*, and a manages b, then the pair $\langle a, b \rangle$ will be an association entity in the *Manager* association, and if $\langle a, b \rangle$ is an association entity in *Manager*, then a manages b.

DEFINITION 6 [7]. *For each parent set S_i, the min/max value of S on S_i is a pair of values (p, q), where p is the minimum and q the maximum number of association entities in S in which any given entity in S_i participates.*

An n-ary association has n min/max values. Possible values for the min are 0 and 1 and for the max are 1 and $\sim\wedge$. A max value of $\sim\wedge$ means 'no upper bound' and a min value of 0 'no lower bound'. A min/max value of $(0, \sim\wedge)$ for the *Manager* association on the left parent set states that an employee may not manage any employees or he may manage any number. For binary associations, a min value of 0 specifies a *partial* or *into* mapping, and a min value of 1 a *total* or *onto* mapping.

DEFINITION 7. *To declare a set is to give it a name and an arity.*

DEFINITION 8. *A SET schema is a collection of declared sets.*

An extensive classification of declared sets is given in [11]. For this paper, three different classes are of interest.

DEFINITION 9. *A primitive set is a set whose members are considered to be indivisible. The arity of a primitive set is zero.*

DEFINITION 10. *A base set is a declared set which cannot be defined in terms of previously declared base or primitive sets. The intension of a base set is, therefore, necessarily expressed informally in a natural language such as English.*

DEFINITION 11. *A defined set is a declared set whose intension can be expressed in the language DEFINE* [11, 12] *for the SET model.*

DEFINITION 12. *Two sets are intensionally equal if they are never extensionally unequal.*

The following conventions will be adopted for naming sets:

(1) Upper case letters are used for naming sets.
(2) Sets that are intensionally equal have the same name.

The following conventions will be used for denoting members of sets:

(1) Lower case letters $a, b, c, \ldots$ denote primitive entities.
(2) The relationship between primitive sets and their members is indicated by corresponding names written in upper and lower case. $a, b, c, \ldots$ are members of $A, B, C, \ldots$, respectively.

The intensions of intensionally equal sets may differ. For example, the intensions "$\{x \in Z : 3 \leq x \leq 5\}$," "$\{3, 4, 5\}$," and "integers between 3 and 5 inclusive" state the same property for membership in a set. The arity of a set is part of its intension. Intensionally equal sets may differ in their arity.

A fundamental difference between certain SET schemas is the conceptualization of an object as primitive in one schema and as a tuple of objects in the other. This observation provides a basis for the definition of a rule for transforming a given schema to a different but equivalent one.

DEFINITION 13. *Schema S_1 is equivalent to schema S_2 iff*

(1) *for every declared set s_1 in S_1, there exists an intensionally equal declared set s_2 in S_2 or s_2 can be declared as a defined set from the sets in S_2.*
(2) *for every declared set s_2 in S_2, there exists an intensionally equal declared set s_1 in S_1 or s_1 can be declared as a defined set from the sets in S_1.*

Since we assume that two sets are intensionally equal if they have the same name, the definition can be restated as follows:

DEFINITION 13′. *Schema S_1 is equivalent to schema S_2 iff, for every declared set in one, there exists a declared set in the other with the same name or a set with that name can be declared as a defined set.*

If two English inputs I_1 and I_2 are equivalent in the sense that they are rephrasings of each other, then *equivalence-preserving* operations on SET-Association diagrams should permit translations between I_1 and I_2. Making use of our previous results in the resolution of ambiguous natural language database NL DB requests [7], we now deal with multiple schemata to gain an understanding of transitions in SET-Association diagrams that are equivalence preserving. Johnson and Rosenberg's heuristic [7] is a generalization of Wald and Sorenson's [6] to include information expressed by the min values of associations in addition to the max values used in the previous work. Johnson and Rosenberg's semantic relatedness measure will be referred to as SRM. We have used SRM as a basis for defining equivalence. Equivalence preserving transition rules will become apparent upon inspection of equivalent schemata.

3. SEMANTIC RELATEDNESS MEASURE

A SET schema is represented by one or more not necessarily connected directed acyclic graphs such as those illustrated in Figures 1 and 2. Association S with parent sets $S_1, \ldots, S_n$ is denoted by $n+1$ vertices labeled $S, S_1, \ldots, S_n$ and n directed edges $(S_1, S), \ldots, (S_n, S)$. Direction on an edge indicates the parentage of sets. An edge (S_i, S) directed from S_i to S indicates that S_i is an immediate parent of S and is labeled with the min/max value of S on S_i. Such a graph is called a *domain graph* [11,12]. We also refer to a domain graph as a *SET-Association diagram.*

Each interpretation of a natural language database NL DB request is represented as a subgraph of the domain graph for a SET schema. The words of an NL DB request denote vertices of the domain graph as in [7]. The set of vertices denoted by words of an NL DB request is called a *target graph.* For a request such as "Which students run programs," if the noun "student" denotes the vertex *Student*, the verb "run" denotes the vertex "Execute" and the noun "program" denotes the vertex *C_program*, then the target graph is {*Student,Execute,C_program*}.

DEFINITION 14. *An NL DB request with target graph TG is applicable to a particular schema, say $Schema_1$, if the domain graph for $Schema_1$ contains TG as a subgraph.*

If the words of an NL DB request denote vertices all of which appear in the domain graph DG for a schema, then the request is applicable to the schema represented by DG. There may be more than one way of connecting the vertices of a target graph.

3.1. Weighted Query Graph—Concept and Related Method

Steiner trees for TG (subtrees of the domain graph that include TG as a subgraph) correspond with possible interpretations for an NL DB request. The graph representations of interpretations are called *query graphs* [6]. Semantic relatedness for an interpretation is measured by computing a directed weight for each of its query graphs with the minimum weight query graph having the greatest semantic relatedness. The method is detailed in [6,7].

In SRM, weights for the edges of a query graph $QG = (V, E)$ are computed from the min/max values that label the corresponding edges of the domain graph. The *forward edges* of QG (with target graph TG) relative to a given vertex $v \in TG$ are those edges $e \in E$ that point away from v. Forward edges relative to v if labeled with min/max (1,1) have a weight of 0 relative to v, if labeled with min/max (0,1) or $(1, \sim\wedge)$ have a weight of 1 relative to v, and if labeled with min/max $(0, \sim\wedge)$ have a weight equal to the cardinality of V. Backward edges relative to v have a weight of zero relative to v independent of their edge labels. Note that an edge labeled with min/max value (1,1) always has a relative weight of zero regardless of its direction. In [7], we

show that the weights assigned to the different types of edges can be varied considerably without affecting the outcome of SRM. That is, even if values other than 1 were assigned to (0,1) and $(1, \sim\wedge)$ edges (say values 5 and 10, respectively) SRM would give the same partial ordering on query graphs for a given target graph.

The weight of QG relative to $v \in TG$ is the sum of the weights of edges $e \in E$ relative to v or more simply the sum of the weights of forward edges relative to v. The weight of QG is the minimum of the relative weights over all $v \in TG$. Query graphs $QG_1, QG_2, \ldots QG_n$ for a given target graph are compared by comparing the absolute weights of $QG_1, QG_2, \ldots QG_n$. If the absolute weight of QG_i is less than the absolute weight of QG_j, $1 \leq i \leq n$, $1 \leq j \leq n$ then the interpretation determined by QG_i is considered to be more likely than the interpretation determined by QG_j.

An example follows to illustrate the concept of query graph and related method for computing semantic relatedness. An ambiguous NL DB request is one with more than one interpretation in the domain graph to which it is applicable. One source of ambiguity arises when a word of the NL DB request has more than one meaning. In this case, there is more than one target graph for a request each of which may determine more than one query graph. Suppose that there are m target graphs for a request. The problem of resolving word sense ambiguity involves, first choosing the minimum weight query graph $QG_{\min}(TG_i)$ for each target graph TG_i, $1 \leq i \leq m$ and second choosing the minimum weight query graph among $QG_{\min}(TG_1), QG_{\min}(TG_2), \ldots, QG_{\min}(TG_m)$.

EXAMPLE 1. Given the request "Which students run programs," assume that the noun "program" can mean either a computer program or a recreational program, and that there are two senses for the verb *run*, one for each sense of the noun "program." A possible domain graph for this situation follows:

$$\begin{array}{c} (1,1) \qquad (0,\sim\wedge) \qquad (0,1) \qquad (0,\sim\wedge) \\ (C\_program \longrightarrow Execute \longleftarrow Student \longrightarrow Administer \longleftarrow R\_program). \end{array}$$

The min/max values state that a computer program is executed by exactly one student, a student executes any number of computer programs, a student administers at most one recreational program, and a recreational program is administered by any number of students.

If the noun "program" denotes the vertices *C_Program* and *R_Program*, the noun "student" denotes vertex *Student*, and the verb "run" denotes vertices *Execute* and *Administer*, then the possible target graphs and their associated query graphs for the request are:

1. TG_1: {*C_program, Execute, Student*}
   QG_1: *C_program* $\longrightarrow$ *Execute* $\longleftarrow$ *Student*
2. TG_2: {*R_program, Administer, Student*}
   QG_2: *Student* $\longrightarrow$ *Administer* $\longleftarrow$ *R_program*
3. TG_3: {*C_Program, Student, Administer*}
   QG_3: *C_program* $\longrightarrow$ *Execute* $\longleftarrow$ *Student* $\longrightarrow$ *Administer*
4. TG_4: {*R_Program, Student, Execute*}
   QG_4: *Execute* $\longleftarrow$ *Student* $\longrightarrow$ *Administer* $\longleftarrow$ *R_program*

QG_3 has a weight greater than or equal to that of QG_1 because QG_1 is a subgraph of QG_3. QG_4 has a weight greater than or equal to that of QG_2 because QG_2 is a subgraph of QG_4. The weights of QG_1 relative to *C_Program*, *Execute*, and *Student* are, respectively, 0, 0, and 3. The weights of QG_2 relative to *R_Program*, *Administer*, and *Student* are, respectively, 3, 0, and 1. The weights of QG_1 and QG_2 are both 0. The weight of QG_3 is 1 (determined by weight relative to *C_program*) and of QG_4 is also 1 (determined by weight relative to *Execute*).

The interpretations "Which students execute recreational programs" and "Which students administer computer programs" are least favored because the weights of the corresponding query graphs (QG_3 and QG_4) are high. An alternative approach to excluding the interpretations

"Which students execute recreational programs" and "Which students administer computer programs" would be the use of selectional restrictions which, in each case, state that the verb does not allow the given type of argument. The remaining query graphs QG_1 and QG_2 have identical weights.

Example 1 has illustrated the use of SRM for resolving word sense ambiguities. The domain involves students, computer programs, and recreational programs. The verb "run" is ambiguous in the domain as is the noun "program." For the request "Which students run programs," SRM does not distinguish the interpretations "Which students execute computer programs" and "Which students administer recreational programs." Context analysis may provide a complementary heuristic to assist in resolving the ambiguity. For example, if "recreational programs" have been previously referenced or more recently referenced in the dialog than "computer programs," then the interpretation "Which students administer recreational programs" would be favored. However, it may be that SRM is giving the correct partial ordering on interpretations without the need for additional heuristics to handle unresolved ambiguity. There may be redundancy in the schema. QG_1 and QG_2 may in fact be equivalent.

4. SENSITIVITY OF SRM TO SET SCHEMA DESIGN ALTERNATIVES

SRM is invariant to arbitrary decisions made of the designer of the schema. Suppose that TG is a target graph for some NL DB request applicable to $Schema_1$ and that $Schema_1$ and $Schema_2$ are equivalent. Augment $Schema_2$ by declaring defined sets for each name labeling a node in TG which does not already appear in $Schema_2$. The outcome of SRM should be the same whether it operates in $Schema_1$ or the augmented version of $Schema_2$. We will show this result (the sufficient condition). It may seem that pairs of schemas related in other ways will also give the same outcome when SRM is applied to them for a given NL DB request. We will show that this is not true for a limited case within a framework that facilitates proof of the more general result.

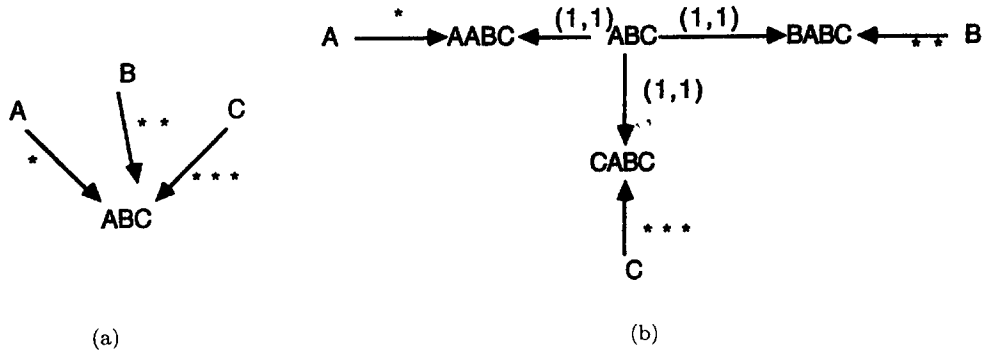

Figure 1. A ternary association expressed as three binary associations.

4.1. The Sufficient Condition

A ternary association expressed as a collection of three binary associations is illustrated in Figure 1. By *expressed*, I mean that schemas (a) and (b) of the figure are equivalent. The extensions of ABC (assumed to be base) of (a) and of ABC (assumed to be primitive) of (b) bear a one-to-one correspondence in the following way: If $\langle a, b, c\rangle \in ABC$ (base), then there is one and only one corresponding object $x \in ABC$ (primitive) where x is the conceptualization of the association entity $\langle a, b, c\rangle$ as an indivisible object. When such a correspondence exists

between two sets with the same name, the sets are said to be related by means of an *entity versus tuple decision.*

THEOREM 1. *An n-ary association can be expressed solely as a collection of n binary associations.*

PROOF. Suppose that $Schema_1$ includes an n-ary association as a base set and that $Schema_2$ is exactly like $Schema_1$ except that the n-ary association is replaced with n binary associations. Since $Schema_1$ and $Schema_2$ are identical with the exception of the n-ary versus n binary associations, to show that $Schema_1$ and $Schema_2$ are equivalent, it suffices to show two things: that the n-ary association can be defined from the n binary associations and that the n binary associations can be defined from the n-ary association.

Without loss of generality, we will consider only ternary associations. Ternary association ABC with parent sets A, B, and C is illustrated in Figure 1a. From the figure, it can be concluded that ABC is either base or defined. ABC cannot be primitive because it has parent sets. ABC of (b) is primitive even though intensionally equal set ABC of (a) is assumed to be base.

The following definition is nonstandard in relational databases and suitable for a synthetic as opposed to decomposition approach to relational database design:

DEFINITION 15 [11]. *Given association ABC with parent sets A, B and C, the projection of ABC on A, (the set AABC) consists of those pairs* $\langle a, \langle a, b, c\rangle\rangle$ *for which* $\langle a, b, c\rangle \in ABC$.

Figure 1b illustrates the projections of ABC of (a) on each of its parent sets. It follows from Definition 15 that the min/max value of $AABC$ on A is the same as that of ABC on A. Edges with identical min/max values are pointed out in the figure by a corresponding numbers of stars on the edges. For example, edges $A \longrightarrow ABC$ and $A \longrightarrow AABC$ have the same min/max values which may be any one of the four possibilities. Since every association entity $\langle a, b, c\rangle \in ABC$ has exactly one A-component, the min/max value of $AABC$ on A is (1,1). Similarly, min/max values of $BABC$ on B and of $CABC$ on C are both (1,1). Observe that, no matter what the extension and min/max values of ABC, the min/max value of $AABC$ on A, of $BABC$ on B and of $CABC$ on C are all (1,1). From Definition 15, there is exactly one a associated by $AABC$ with $\langle a, b, c\rangle \in ABC$ since a typical member of $AABC$ looks like $\langle a, \langle a, b, c\rangle\rangle$.

Definition 13′ of equivalent schemas was motivated by the following argument: There is a one-to-one relationship between ABC (base) and ABC (primitive) sufficient to make schemas (a) and (b) equivalent. There is also a one-to-one relationship between ABC (base) and $AABC$ but not sufficient to make the two schemas equivalent. Schemas (a) and (b) are equivalent because first, for every set in (a), there is a set with the same name in (b) and second, sets of (b) with names $AABC$, $CABC$ and $BABC$ can be defined in terms of primitive and base sets of (a). ∎

Now that it has been established that schemas (a) and (b) of Figure 1 are equivalent, let us consider the outcome of our heuristic SRM [7] for measuring semantic relatedness in each schema. Referring again to Figure 1, since the weight of a (1,1) edge is 0, the weights of the edges $ABC \longrightarrow AABC$, $ABC \longrightarrow BABC$, and $ABC \longrightarrow CABC$ are all zero. If the target graph for a request is $TG = \{A, B, C\}$ then, although the query graphs for TG in the two schemas are different, their weights are the same.

THEOREM 2. *A sufficient condition for a natural language database request to be applicable to* $Schema_1$ *and* $Schema_2$ *is that the schemas are obtained from each other by entity versus tuple transformations.*

PROOF. No additional nodes or edges are introduced in the transformation and no other edges or nodes are removed. Therefore, if a given request is applicable in one schema, it will be applicable in the other. ∎

THEOREM 3. *A sufficient condition for SRM to give the same ordering on interpretations when applied to a natural language database request applicable to* $Schema_1$ *and* $Schema_2$ *is that the schemas are obtained from each other by entity versus tuple transformations.*

PROOF. Only (1,1) edges are introduced in transformations involving entity versus tuple decisions. The weight of a (1,1) edge is zero. If such an edge appears in a query graph, it adds nothing to the weight of the query graph. Therefore, the relative ordering of interpretations provided by our semantic relatedness heuristic will be the same in the two schemas. ∎

THEOREM 4. *A sufficient condition for SRM to give the same ordering on interpretations when applied to a natural language request applicable to* $Schema_1$ *and* $Schema_2$ *is that the schemas are equivalent.*

PROOF. There are no ways for equivalent SET schemas to differ but for intensionally equal sets to have different arity. ∎

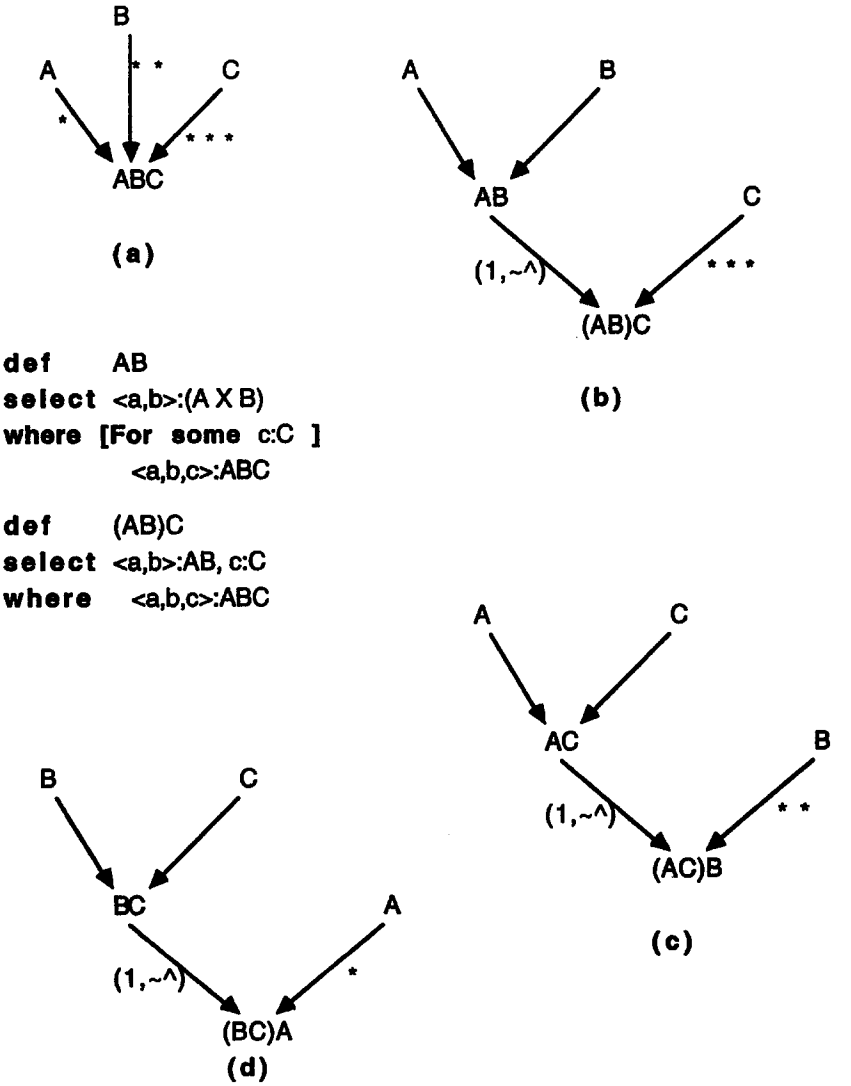

Figure 2. A ternary association expressed as a collection of binary associations.

4.2. The Necessary Condition

I propose that parallelism in SET schemas is necessary to permit all pairs of schemas obtained from each other by means of entity versus tuple decisions to be equivalent. In this section, a limited result for necessity is proved which may be extendable and has to be investigated further.

Consider schema (a) of Figure 2 and the definition in the language DEFINE [11,12] of the sets AB and $(AB)C$. In DEFINE, the semicolon ":" is used in place of $\in$. A pair $\langle a, b\rangle$ cannot be a member of AB (defined) without also being the left element of a pair $\langle\langle a, b\rangle, c\rangle$ of $(AB)C$

(defined). Therefore, the min value of $(AB)C$ (defined) on AB is 1. Consider $(AB)C$ (base) of schema (b). Since sets with the same name are assumed to be intensionally equal, the min value of $(AB)C$ (base) on AB is also 1. If the max value of at least one of ABC on A and on B is 1, then the max value of $(AB)C$ (base) on AB is 1. Since the min/max values of ABC on A and on B are unspecified, the max value of $(AB)C$ (base) on AB is $\sim\wedge$.

Consider schemas (c) and (d). The definitions of AC and BC are obtained by interchanging names of parent sets in the definition of AB. To get the definition of AC for example, swap the names A and C. The definitions of $(AC)B$ and $(BC)A$ are obtained by replacing names in the definition of $(AB)C$. For example, the definition of $(AC)B$ is obtained by replacing the name AB with AC and the name C with B. By similar arguments, the min/max values of $(AC)B$ on AC and of $(BC)A$ on BC are both $(1, \sim\wedge)$.

The min/max values of ABC on C and of $(AB)C$ on C are identical since whenever $\langle a, b, c\rangle \in ABC$, $\langle\langle a, b\rangle, c\rangle \in (AB)C$. The min value of ABC on A is equal to the min value of AB on A, since, whenever $a \in A$ participates in association entity $\langle a, b, c\rangle$ of ABC, it also participates in a pair $\langle a, b\rangle$ of AB.

The max value of ABC on A and of AB on A are not necessarily equal. If the max value of ABC on A is 1, then the max value of AB on A is also 1, but not the converse. If the max value of ABC on A is $\sim\wedge$, then the max value of AB on A may be 1.

Notice that for each of the domain graph representations of schemas (b), (c), and (d) one of the vertices A, B, or C is distinguished as having at most one forward edge relative to it. Vertex C is so distinguished in (b), and we will refer to an edge such as $C \longrightarrow (AB)C$ as a *side edge.*

The min/max values of ABC on A, on B, and on C are identical to the min/max values on the side edges of (d), (c), and (b), respectively, as indicated by the stars in the figure. It is not possible to define a set on any one of (b), (c), or (d) that is intensionally equal to ABC of schema (a).

```
def      ABC
select   a:A,b:B,c:C
where    <<a,b>,c>:(AB)C
```

Figure 3. Incorrect definition of ABC from Schema (b).

Consider the incorrect definition of ABC given in Figure 3. The min/max value of ABC on C is equal to the min/max value of $(AB)C$ on C. The min value of ABC on A (on B) is equal to that of AB on A (on B). However, the max value of ABC on A (on B) may differ from that of AB on A (on B). A counter example follows: If the max value of AB on A is 1 and of $(AB)C$ on C is $\sim\wedge$, then the max value of ABC on A may be 1. If the max value of ABC on A is $\sim\wedge$, then a possible extension for AB is $\{\langle a, b\rangle\}$ and for $(AB)C$ is $\{\langle\langle a, b\rangle, c_1\rangle, \langle\langle a, b\rangle, c_2\rangle\}$. By the definition of ABC, the extension of ABC is $\{\langle a, b, c_1\rangle, \langle a, b, c_2\rangle\}$ and, hence, the max value of ABC on A is not equal to 1.

One way to construct a schema that expresses the same information as (a) is to combine schemas (b), (c), and (d). Figure 4 gives a definition of the set ABC of schema (a) from schemas (b), (c), and (d). The definition guarantees that the min/max values of the defined set ABC are identical to the corresponding ones of the base set ABC.

The *union* of two domain graphs is the set of vertices and edges (together with edge labels) that appear in one or other or both of the graphs. Let (b + c + d) denote the schema which is the union of schemas (b), (c), and (d).

By Definition 13′, schemas (a) and (b + c + d) are equivalent. Intensions of sets, whose names label nodes in (b + c + d) but not (a), are expressed by the DEFINE statements of Figure 2 (and DEFINE statements that can be obtained from the given ones by set name substitutions). The intensions of sets named in (a) but not (b + c + d) (only the set ABC) is given in Figure 4.

```
def      ABC
select   a:A,b:B,c:C
where    <<a,b>,c>:(AB)C
  and    <<a,c>,b>:(AC)B
  and    <<b,c>,a>:(BC)A
```

Figure 4. Correct Definition of ABC from Schema (b+c+d).

Equivalent schemas should give identical semantic relatedness for a given NL DB request applicable in each one. Each of (b), (c), and (d) is a query graph in $(b+c+d)$ for a request that references vertices A, B, and C. The following observations leads us to conclude that the weights of the minimum weight query graphs for target graph $TG = \{A, B, C\}$ in (a) and in $(b+c+d)$ are identical.

RESULT 1. *The minimum weight query graph among (b), (c), and (d) has weight equal to the weight of its side edge.*

PROOF (BY CONTRADICTION). Suppose that (d) is the minimum weight query graph among (b), (c) and (d). Suppose, contrary to Result 1, that the weight of (d) is determined by vertex C. There are two forward edges relative to C: $C \longrightarrow BC$ and $BC \longrightarrow (BC)A$. The min/max value for $C \longrightarrow BC$ is not specified in the figure. Let us assume that it is a $(0, \sim\wedge)$-edge of weight greater or equal to the weight of any other edge. $BC \longrightarrow (BC)A$ is a $(1, \sim\wedge)$-edge of weight 1. If the weight of a $(0, \sim\wedge)$-edge is w, then the weight of (d) is $w + 1$.

There is another query graph (b) whose weight is determined by the side edge $C \longrightarrow (AB)C$. The weight of $C \longrightarrow (AB)C$ and also of (b) is smaller or equal to w. Since the weight of (d) is $(w + 1)$, the weight of (b) is smaller than the weight of (d), which contradicts the assumption that the minimum weight query graph is (d). ∎

RESULT 2. *The weight of (b), (c) or (d) is equal to the weight of its side edge.*

PROOF. Since (b), (c), and (d) are the same graphs but with the vertices relabeled, we could, in the proof of Result 1, equally well have assumed that either (b) or (c) is the minimum weight graph and derived a contradiction by the same argument. Hence, independent of which of (b), (c), or (d) is the minimum weight query graph, its weight will be equal to the weight of its side edge. ∎

RESULT 3. *The minimum weight query graph among (a), (b), (c), and (d) is (a).*

PROOF. Suppose that weight of (b) is strictly less than weight of (a). The side edge $C \longrightarrow (AB)C$ of (b) is labeled with the same min/max value as the edge $C \longrightarrow ABC$ of (a). (Observe that there are three stars labeling both edges.) Without loss of generality, suppose that the weight of $B \longrightarrow ABC$ of (a) is strictly less than the weight of $C \longrightarrow ABC$ of (a) and that weight of (a) is determined by vertex B. The side edge $B \longrightarrow (AC)B$ of (c) is labeled with the same min/max value as the edge $B \longrightarrow ABC$ of (a). (Each edge is labeled with two stars.) From Result 2, query graph (c) has weight equal to the weight of its side edge. Therefore, the weights of (a) and (c) are identical. Also from Result 2, the weight of (b) is equal to the weight of its side edge. The side edge $B \longrightarrow (AC)B$ of (c) has weight strictly less than the side edge $C \longrightarrow (AB)C$ of (b). Therefore, the weight of (c) is strictly less than the weight of (b). Let a denote the weight of (a). We have assumed $b < a$ and shown $a = c$ and $c < b$ from which we can conclude $b < b$. Our assumption must be wrong that the weight of (b) is strictly less than weight of (a). Since (b), (c), and (d) are the same graphs but with the vertices relabeled, we can conclude by the same argument that neither (c) nor (d) has weight strictly less than weight of (a). Therefore, (a) is the minimum weight query graph among (a), (b), (c), and (d). ∎

RESULT 4. *Whichever vertex, say V, determines the weight of (a), it is the query graph QG with side edge starting at V that is minimum weight among (b), (c) and (d).*

PROOF. Suppose that weight of (a) is determined by vertex C. Therefore, the weights of edges $A \longrightarrow ABC$ and $B \longrightarrow ABC$ of (a) are greater than or equal to the weight of $C \longrightarrow ABC$. Since the weight of (a) is equal to the weight of $C \longrightarrow ABC$, and the weight of (b) is equal to the weight of the side edge $C \longrightarrow (AB)C$, and the min/max values of $C \longrightarrow ABC$ and $C \longrightarrow (AB)C$ are identical, it follows that the weights of (a) and (b) are identical.

We could have assumed that weight of (a) is determined by one of the other vertices in TG and derived by the same argument that weight of (a) is equal to weight of QG with side edge starting at that vertex. From Result 3, (a) is minimum weight among (a), (b), (c) and (d). Therefore, QG with the relevant side edge will be minimum weight among (b), (c) and (d). ∎

It follows from Results 1 and 3 that semantic relatedness for target graph $TG = \{A, B, C\}$ in (a) and in (b + c + d) are identical. Any one of schemas (b), (c) or (d) alone cannot be considered as a possible design for a schema to which TG is applicable. Semantic relatedness for TG in schema (a) and in schemas (b), (c) or (d) may differ. If one of (b), (c) or (d) alone is admitted as a possible design for a schema to which TG is applicable, then all three must be admitted. The schema designer will not have knowledge which will permit a choice among the schemas (b), (c) and (d) even though semantic relatedness for QG (a) and at least one of QG's (b), (c) and (d) are identical. Therefore, (b + c + d) is the only schema which can be considered as an alternative to (a) if SRM is to give the same outcome in both schemas. It remains to be shown that (a) and (b + c + d) can be obtained from each other by entity versus tuple transformations.

In this section, the properties of four specific schemas are proved. The main result is that parallelism (the ability to duplicate parts of the graph save for the renaming of vertices) is necessary to permit a ternary association to be expressed solely as a collection of binary associations. The proof needs to be generalized to n-ary associations, $n > 3$. That is, it remains to show that parallelism is necessary to express an n-ary association as a collection of associations of arity less than n. Parallelism is necessary to express equivalence and a notion of equivalence provides the conceptual foundations needed to express the notion of semantic relatedness.

5. APPLICATIONS

In this section, we show how the results of Section 4 can be applied. Consider SET schemas S_{music} and S_{baseball} for the music and baseball domains, respectively. Consider natural language database requests $\text{NLDB}_{\text{music}}$ and $\text{NLDB}_{\text{baseball}}$ applicable to S_{music} and S_{baseball}, respectively. To say that $\text{NLDB}_{\text{music}}$ is applicable to S_{music} is to say that the target graph for $\text{NLDB}_{\text{music}}$ is a subgraph of S_{music}. The strength of a relationship between concepts referred to in $\text{NLDB}_{\text{music}}$ is *apparent* in S_{music} only with respect to a schema in which a comparison can be made. If $\text{NLDB}_{\text{music}}$ is applicable to both schemas then a relative measure of the strength of the relationship can be obtained by applying SRM to $\text{NLDB}_{\text{music}}$ in the union $(S_{\text{music}} + S_{\text{baseball}})$. If $\text{NLDB}_{\text{music}}$ is applicable to both schemas it will be applicable to the union.

Theorems 2 and 3 give sufficient conditions, defined in terms of entity versus tuple transformations, for SRM to give the same relative partial ordering on interpretations for a given request in two different schemas. Suppose that SET schema $S_{\text{lullabies}}$ is equivalent to S_{music}. A natural language database request if applicable to S_{music} will also be applicable to $S_{\text{lullabies}}$. Suppose that $\text{NLDB}_{\text{music}}$ references the concepts of *tone* and *pitch*. If the directness of the relationship between those concepts is apparent in S_{music}, then it will also be apparent in $S_{\text{lullabies}}$. The synonyms "pitch," "quality," and "strength," if applicable to S_{music} will also be applicable to $S_{\text{lullabies}}$. The concepts denoted by those nouns, if strongly related in S_{music}, will also be strongly related in $S_{\text{lullabies}}$.

Suppose that S_{diamond} is equivalent to S_{baseball}. The weak relationship between the concepts of *pitch* and *tone* if apparent in S_{baseball} will also be apparent in S_{diamond}. The synonyms "pitch,"

"quality," and "strength," if applicable to S_{baseball} will also be applicable to S_{diamond}. The corresponding concepts if weakly related in S_{baseball} will also be weakly related in S_{diamond}.

Schema equivalence is expressed in terms of entity versus tuple transformations. Theorem 4 states that, if S_{music} and S_{baseball} are equivalent schemas, then SRM gives the same relative partial ordering on interpretations independent of which schema is used to express the domain. The theorem states a desirable property of a semantic relatedness measure, one that is not shared by previous metrics [1] based on counting the number of links in the database navigational path that represents the meaning of the request.

The results of Section 4.2 give necessary conditions, defined in terms of entity versus tuple transformations, for SRM to give the same relative partial ordering on interpretations for a given request in two different schemas. Suppose that S_{music} expresses a ternary association (as in Figure 2a), S_{baseball} expresses a collection of binary associations (as in Figure 2b), and $\text{NLDB}_{\text{music}}$ is applicable to both schemas. Our fundamental premise is that, if SRM does not give the same partial ordering on interpretations for $\text{NLDB}_{\text{music}}$ in S_{music} and S_{baseball}, then the two schemas are not equivalent. To give the same partial ordering on interpretations for $\text{NLDB}_{\text{music}}$ involving the concepts of *pitch* and *tone*, for example, S_{baseball} would have to be augmented with additional paths of concepts connecting the concepts of *pitch* and *tone*. Suppose that Figure 2 can be generalized to express an n-ary association as a collection of associations of arity less than n. We argue that SRM gives the same partial ordering on interpretations for $\text{NLDB}_{\text{music}}$ only in schemas which are obtained from S_{music} by entity versus tuple transformations. Our argument is based on the observation that Figure 2 (generalized to include n-ary associations, $n > 3$) captures all of the ways in which equivalent SET schemas can differ.

6. CONCLUSIONS

The minimal arbitrariness in the structure of SET schemas is an expression of the natural concept of concurrency. Symmetric SET schemas differ from each other in the renaming of vertices. In particular, the primitive sets are renamed. This forces a renaming of the base sets. Parallel SET schemas are unions of symmetric SET schemas. Parallelism is necessary to ensure that all pairs of schemas that are related to each other by the entities in one being considered as association entities in the other are equivalent. Parallelism is part of the foundations needed to express semantic relatedness.

The only structural differences between semantically equivalent SET schemas are those that arise from entity versus tuple decisions. Semantic relatedness measures depend on the structure of schemas. There are no structural components of the SET schema that do not reflect the semantics of the domain. Our metric for semantic relatedness is invariant to arbitrary decisions made by the designer of the schema. The only arbitrariness in the design arises from entity versus tuple decisions and we admit such differences in our definition of equivalent SET schemas by permitting intensionally equal sets to differ in their arity.

Fundamental principles for designing SET schemas are the following:

(1) A set that can be declared as a defined set should never itself be declared as a base set.
(2) An entity that we wish to conceive of as primitive should be treated that way. It should never be treated as a tuple.

In this paper, we have assumed (1) and investigated (2). Our definition of equivalence of SET schemas is a restatement of principle (2). If SET schemas are designed according to principle (1), then there are no differences between equivalent SET schemas other than those arising from entity versus tuple decisions.

Semantic relatedness for parallel representation of different interpretations of the same natural language input has been addressed in this research. The notions of n-ary set and min/max value permit a definition of equivalence for conceptual modeling for the design of a natural language

interface. The intuitive notion of semantic relatedness is completely formalized in terms of the concepts of n-ary set and min/max value.

A notation with algebraic properties is provided by our framework. Objects in the algebra are domain graphs and operations on domain graphs permit entity versus tuple transformations. With these conceptual foundations, we will be able to translate between intensions for intensionally equal sets. For intensions that are formal statements in a language such as DEFINE, the problem involves mapping a DEFINE query to a subgraph of the domain graph. For intensions that are expressed informally in natural language, the problem involves translating between internal representations of alternate rephrasings of the same English input.

REFERENCES

1. S.J. Kaplan, Designing a portable natural language database query system, *ACM Transactions on Database Systems* **9** (1), 1–19 (March 1984).
2. G. Hall, W.S. Luk, N. Cercone and P. McFetridge, A solution to the map problem in natural language interface construction, In *Proceedings of International Computer Science Conference*, pp. 351–359, The Computer Society of the IEEE, Hong Kong Chapter, (1988).
3. D. Lin, Automatic logical navigation among relations using Steiner trees, Presented at the 5^{th} *International Conference on Data Engineering*, pp. 582–588, IEEE, Los Angeles, (1989).
4. A. Pahwa and A. Arora, Automatic database navigation: Towards a high level user interface, In *Proceedings of Conference on the Entity-Relationship Model*, pp. 36–43, (1985).
5. A. Motro, Constructing queries from tokens, In *Proceedings of SIGMOD '86 Conference*, pp. 120–131, (1986).
6. J.A. Wald and P.G. Sorenson, Resolving the query inference problem using Steiner trees, *ACM Transactions on Database Systems* **9** (3), 348–368 (September 1984).
7. J.A. Johnson and R.S. Rosenberg, A measure of semantic relatedness for resolving ambiguities in natural language database requests, *Data & Knowledge Engineering* **7** (3), 201–225 (1992).
8. J. Bear and J.R. Hobbs, Localizing expression of ambiguity, Presented at the *Second Conference on Applied Natural Language Processing*, pp. 235–241, (1988).
9. M.S. Schmalz, Knowledge structures and algorithms for the parallel processing of natural language in the image domain, In *Proceedings of the Fifth Florida Artificial Intelligence Research Symposium*, Florida AI Research Society, (1992).
10. G.X. Ritter, J.N. Wilson and J.L. Davidson, Image algebra: An overview, *Computer Vision, Graphics, and Image Processing* **49**, 297–331 (1990).
11. P.C. Gilmore, Concepts and methods for database design, Technical Report TR87-31, Department of Computer Science, University of British Columbia, (1987).
12. P.C. Gilmore, A foundation for the Entity Relationship approach: How and why?, In *Proceedings of the* 7^{th} *International Conference on Entity-Relationship Approach*, New York, (1987).
13. P.C. Gilmore, Natural deduction based set theories: A new resolution of the old paradoxes, *J. Symb. Logic* **51** (2) (1986).
14. W. Kent, Fact-based data analysis and design, In *Entity-Relationship Approach to Software Engineering*, (Edited by C.G. Davis, S. Jajodia, P.A. Ng and R.T. Yeh), pp. 3–53, Elsevier Science Publishers B.V. North-Holland, (1983).